

Using EPICS on Nova with the RFIES Development Tools

Support for EPICS software development has been added to the RFIES software development tools (ESD tools.) On the IOC side, there is a tool for creating an EPICS_TOP directory and tools for creating IOC and IOC support projects. On the client application side, there is a tool for creating a MY_TOP directory and another for creating client application program projects. A few of the original ESD tools are of little or no use to EPICS developers, but in all there is enough support to get you going with EPICS in a nearly painless manner. The EPICS support tools are designed to work with the EPICS installation (currently 3.14.10) on nova.fnal.gov as provided by the Accelerator Controls Department.

This note is not an ESD tools or EPICS tutorial¹ -- it assumes a basic familiarity with the concepts and nomenclature of the ESD tools and EPICS. Here we attempt to introduce new tools supporting EPICS development, list some of the original ESD tools that remain useful with EPICS and point toward some useful IOC examples.

To begin one needs to log in to nova and initialize the ESD tools as usual.

IOC Applications: The newepicstop Tool

The first thing that an EPICS developer needs is a standard EPICS_TOP directory that will contain all IOC applications. The **newepicstop** tool will automatically do this for you:

```
Nova-> newepicstop
```

```
...
```

```
Nova-> pwd
```

```
/export/home/voy/esd/src/epics
```

```
Nova-> ls -al
```

-rw-rw-r--	1	voy	bdrfinst	467	Jun	3 08:43	Makefile
-rw-rw-r--	1	voy	bdrfinst	211	Jun	3 08:43	Targets
drwxrwxr-x	3	voy	bdrfinst	3	Jun	3 08:43	bin
drwxrwxr-x	4	voy	bdrfinst	12	Jun	3 08:43	configure
drwxrwxr-x	2	voy	bdrfinst	7	Jun	3 08:43	db
drwxrwxr-x	2	voy	bdrfinst	5	Jun	3 08:43	dbd

¹ For information on the RFIES Development Tools see: Beams Document 1271-v6 "RF & Instrumentation Front-end Software Development Tools"

For information on EPICS see "www.aps.anl.gov/epics/docs/index.php"

```

drwxrwxr-x  2 voy      bdrfi nst      3 Jun  3 08:43 include
drwxrwxr-x  3 voy      bdrfi nst      3 Jun  3 08:43 lib
drwxrwxr-x  4 voy      bdrfi nst      5 Jun  3 08:43 myexampl eApp
-rwxrwxr-x  1 voy      bdrfi nst     241 Jun  3 08:43 setup-epi cs

```

The **newepicstop** tool builds an EPICS_TOP directory in your tools sandbox area with the path ~/esd/src/epics and then cd's to that location and initializes the EPICS development environment. Essentially we have created a new *special* project called 'epics' in your sandbox. The process of creating an EPICS_TOP is complicated and the **newepicstop** tool will produce a lot of mostly useless output as it proceeds. As you can see, several files and directories are created within the epics sandbox directory. The setup-epics file is the only tools related file and is used to dynamically modify the tools' sandbox base directory from ~/esd/src to ~/esd/src/epics. At this point you are in the EPICS development context and (because your sandbox directory has been changed) your normal ESD projects are no longer available in this login session. *This is important – to get back to normal ESD development you must exit this shell instance (i.e., logout and login again.)* Note that the **newepicstop** tool needs to be invoked only once to create your EPICS_TOP.

At this point there are two really useful commands for working with IOC projects:

```

make realclean – recursively remove all generated files from the cwd
emake – make all IOC targets (currently vxWorks-ppc604_al tivec)

```

The EPICS documentation talks about the need to use gmake for all makes. That will work. The tools create an alias for 'gmake' named **make** for compatibility with the original tools, and another convenience alias for 'gmake targetName' called **emake**. Most EPICS work can be done with **make** and **emake**. You may issue these make commands at any level of the directory structure containing a Makefile -- the EPICS make system is fully recursive.

You may notice an .../epics/myexampleApp directory containing the standard EPICS example IOC application. You may cd into myexampleApp and do an emake to see what happens.

IOC Applications: The newepicsioc Tool

The next thing an EPICS developer is likely to want is an IOC application. The **newepicsioc** tool will automatically create one for you:

```

nova-> newepi cs ioc      hi nsbpm
-t-> Creating new EPICS IOC hi nsbpmApp
-t-> Setting up project hi nsbpmApp ...
-t-> Running /usr/local/vxworks/scripts/wind6.4_mv5500.bash for target
MVE5500_64 ...
nova-> pwd
/export/home/voy/esd/src/epi cs/hi nsbpmApp
nova-> ls -al
drwxrwxr-x  2 voy      bdrfi nst      3 Jun  3 09:38 Db
-rw-rw-r--  1 voy      bdrfi nst      304 Jun  3 09:38 Makefile
-rw-rw-r--  1 voy      bdrfi nst      211 Jun  3 09:38 Targets
drwxrwxr-x  2 voy      bdrfi nst        4 Jun  3 09:38 src
nova-> ls -al src
-rw-rw-r--  1 voy      bdrfi nst     1088 Jun  3 09:38 Makefile
-rw-rw-r--  1 voy      bdrfi nst      407 Jun  3 09:38 hi nsbpmMai n. cpp

```

The **newepicsioc** tool creates a stub IOC project with App appended to the name that you provided, and then cd's to the newly created project directory. The App suffix is enforced by EPICS – you have to learn to live with ‘yourname**App**’! You may do an emake to see what happens.

IOC Applications: The newepicssupport Tool

Another thing an EPICS developer may want is a support application. The **newepicssupport** tool will automatically create one for you:

```

nova-> newepi cssupport  adcdri ver
-t-> Creating new EPICS IOC support library adcdriverApp
-t-> Setting up project adcdriverApp ...
-t-> Running /usr/local/vxworks/scripts/wind6.4_mv5500.bash for target
MVE5500_64 ...
nova-> pwd
/export/home/voy/esd/src/epi cs/adcdri verApp
nova-> ls -al
drwxrwxr-x  2 voy      bdrfi nst      3 Jun  3 09:57 Db
-rw-rw-r--  1 voy      bdrfi nst      303 Jun  3 09:57 Makefile
-rw-rw-r--  1 voy      bdrfi nst      211 Jun  3 09:57 Targets
drwxrwxr-x  2 voy      bdrfi nst        4 Jun  3 09:57 src
nova-> ls -al src
-rw-rw-r--  1 voy      bdrfi nst      660 Jun  3 09:57 Makefile
-rw-rw-r--  1 voy      bdrfi nst      166 Jun  3 09:57 adcdri ver. dbd

```

The **newepicssupport** tool creates a stub support project with App appended to the name that you provided, and then cd's to the newly created project directory. The App suffix is enforced by EPICS – you have to learn to live with 'yourname**App**'! You may do an emake to see what happens.

IOC Applications: Using the setup Tool

The **setup** tool remains useful for doing IOC development but in a somewhat limited manner. Once you have created the special epics project with **newepicstop** you activate it with the '**setup epics**' command:

```
nova-> esd
Running ESD setup ...
nova-> setup epics
-t-> Setting up project epics ...
-t-> Switching to sandbox /export/home/voy/esd/src/epics
-t-> Running /usr/local/vxworks/scripts/wind6.4_mv5500.bash for target
MME5500_64 ...
nova-> pwd
/export/home/voy/esd/src/epics
```

At this point you are in the EPICS development context and your normal ESD projects are no longer available in this login session. *This is important – to get back to normal ESD development you must exit this shell instance (i.e., logout and login again.)*

Note that you cannot create a new EPICS project using the setup tool the way you do for regular ESD projects. For that you must use the **newepicsioc** or **newepicssupport** tool. Once you have an EPICS IOC or support project you can use the '**setup projectName**' command:

```
nova-> setup epics
nova-> setup hinsbpmApp
-t-> Setting up project hinsbpmApp...
-t-> Running /usr/local/vxworks/scripts/wind6.4_mv5500.bash for target
MME5500_64 ...
nova-> pwd
/export/home/voy/esd/src/epics/hinsbpmApp
```

The '**setup epics**' command transitions to the special epics project which is the EPICS development environment. The '**setup hinsbpmApp**' command moves you into the

hinsbpmApp project. The setup tool may be used to transition from the normal ESD context to the EPICS context, and then to move to -- and between -- EPICS IOC projects.

IOC Applications: Other Useful ESD Tools

The following list contains tools that may be useful while doing EPICS IOC development. Items marked with a bullet were added or modified specifically for use with EPICS. The non-bulleted tools retain their normal ESD functionality and might be useful. Generally, the concept of targets and references does not apply to EPICS development so tools like target, references, production, test and development are not useful.

- dumpesd - list all development tool related environment variables
- emake - do a 'gmake vxWorks-ppc604_altivec' for EPICS
- epush - run a script in the current working directory named 'epush'
- finddsp fileName - find specified file in DSP directories (dsp, testdsp and devdsp)
- findepics fileName - find specified file in EPICS include directories
- findinc fileName - find specified file in include directories (inc, testinc and devinc)
- findlib fileName - find specified file in library directories (lib, testlib and devlib)
- findproducts fileName - find specified file in PRODUCTS_INCDIR
- findsrc fileName - find specified file in user's sandbox directory
- findvx fileName - find specified file in VxWorks directory structure
- helpme - list help (this list) for ESD tools
- incvs pattern - check to see if 'pattern' is part of any project name in the CVS repository
- listallp [orgName] - list all projects in the repository and/or the user's sandbox
- listp [orgName] - list all projects in the repository
- lists - list all projects in the user's sandbox
- pull fileName - copy file from the install node:directory (currently nova:/fecode-bd/vxworks_boot/fe)
- push fileName - copy file to the install node:directory (currently nova:/fecode-bd/vxworks_boot/fe)
- readme - do a more of any README file in the cwd
- search searchPattern [dsp|fe|shared|all] - search source repositories for definition of searchPattern
- searchepics fileName - search EPICS src directories for (grep style) searchPattern
- searchheaders searchPattern [inc|testinc|devinc] - search ESD headers for (grep style) searchPattern

searchproducts searchPattern - search PRODUCTS_INCDIR headers for (grep style) searchPattern
searchsrc searchPattern - search user's sandbox directory for (grep style) searchPattern
searchvx searchPattern - search VxWorks headers for (grep style) searchPattern

IOC Applications: An Example IOC Application

The hinst1App project in CVS is a good example of an IOC with several record types and that also uses support libraries that were developed using the standard ESD tool set.

```
nova-> esd
Running ESD setup ...
nova-> setup epics
-t-> Setting up project epics ...
switching to sandbox /export/home/voy/esd/src/epics
-t-> Running /usr/local/vxworks/scripts/wind6.4_mv5500.bash for target
MME5500_64 ...
nova-> setup hinst1App
-t-> Setting up project hinst1App ...
The project hinst1App is not in your sandbox ...
Do you want to check hinst1App out of the repository [ y|n ]? y
cvs checkout: Updating hinst1App
...
nova-> pwd
/export/home/voy/esd/src/epics/hinst1App
nova-> ls -al
drwxrwxr-x  2 voy      bdrfinst      5 Jun  3 12:35 CVS
drwxrwxr-x  3 voy      bdrfinst     27 Jun  3 12:35 Db
-rw-rw-r--  1 voy      bdrfinst    304 Feb 25  2009 Makefile
-rwxrwxr-x  1 voy      bdrfinst    255 Jun  2 08:17 Targets
drwxrwxr-x  3 voy      bdrfinst     19 Jun  3 12:35 src
```

IOC Applications: An Example IOC Support Application

The sineRecordApp project in CVS is a good example of an IOC support project containing a single record that will generate a sine wave. The project is of value because the main source module sineRecord.cpp is well commented to document the relationship between the various record functions and the text entered into sineRecord.db.

```
nova-> esd
Running ESD setup ...
nova-> setup epics
-t-> Setting up project epics ...
```

```

-t-> Switching to sandbox /export/home/voy/esd/src/epics
-t-> Running /usr/local/vxworks/scripts/wind6.4_mv5500.bash for target
MVME5500_64 ...
nova-> setup sineRecordApp
-t-> Setting up project sineRecordApp ...
The project sineRecordApp is not in your sandbox ...
Do you want to check sineRecordApp out of the repository [ y|n ]? y
cvs checkout: Updating sineRecordApp
...
nova-> pwd
/export/home/voy/esd/src/epics/sineRecordApp
nova-> ls -al
drwxrwxr-x  2 voy      bdrfi nst      5 Jun 10 09:59 CVS
drwxrwxr-x  3 voy      bdrfi nst      4 Jun 10 09:59 Db
-rw-rw-r--  1 voy      bdrfi nst    303 Jun 10 09:56 Makefile
-rw-rw-r--  1 voy      bdrfi nst    259 Jun 10 09:56 Targets
drwxrwxr-x  3 voy      bdrfi nst      7 Jun 10 09:59 src
nova-> ls -al src
drwxrwxr-x  2 voy      bdrfi nst      5 Jun 10 09:59 CVS
-rw-rw-r--  1 voy      bdrfi nst    677 Jun 10 09:56 Makefile
-rw-rw-r--  1 voy      bdrfi nst   9624 Jun 10 09:56 sineRecord.cpp
-rw-rw-r--  1 voy      bdrfi nst    177 Jun 10 09:56 sineRecord.db
-rw-rw-r--  1 voy      bdrfi nst   2944 Jun 10 09:56 sineRecord.dbd

```

Client Applications: The newepicsmytop Tool

If you want to develop EPICS client applications you will need a standard MY_TOP directory that will contain all client applications. The **newepicsmytop** tool will automatically do this for you:

```
Nova-> newepicsmytop
```

...

```

Nova-> pwd
/export/home/voy/epicsapps
nova-> ls -al
-rw-rw-r--  1 voy      bdrfi nst    467 Jun  3 11:33 Makefile
drwxrwxr-x  3 voy      bdrfi nst      3 Jun  3 11:34 bin
drwxrwxr-x  6 voy      bdrfi nst    14 Jun  3 11:33 configure
drwxrwxr-x  6 voy      bdrfi nst      9 Jun  3 11:34 exampleApp

```

The **newepicsmytop** tool builds a MY_TOP directory in your home directory with the path ~/epicsapps and then cd's to that location and initializes the EPICS development environment. The MY_TOP creation process is complicated and the **newepicsmytop**

tool will produce a lot of mostly useless output as it proceeds. As you can see several files and directories are created within the epicsapps directory.

At this point you are within the pure EPICS application development regime so the ESD tools are of little value (you can use the newepicsapp tool described below.) There are two really useful commands for working with client projects:

```
gmake realclean – recursively remove all generated files from the cwd
gmake – make all host targets (currently solaris-sparc)
```

Since we are out of the ESD tools regime there are no aliases for gmake – you must use gmake for all client application development. You may issue these gmake commands at any level of the directory structure containing a Makefile -- the EPICS make system is fully recursive.

You may notice an .../epicsapps /exampleApp directory containing the standard EPICS example application. You may cd into exampleApp and do a gmake to see what happens.

Client Applications: The newepicsapp Tool

The next thing an EPICS developer is likely to want is a client application. The **newepicsapp** tool will automatically create one for you:

```
nova-> newepicsapp hi nsDi spl ay
-t-> Creating new EPICS user application hi nsDi spl ayApp
nova-> pwd
/export/home/voy/epicsapps/hi nsDi spl ayApp
nova-> ls -al
-rw-rw-r-- 1 voy bdrfi nst 450 Jun 3 11:52 Makefile
-rw-rw-r-- 1 voy bdrfi nst 661 Jun 3 11:52 caExample.c
-rw-rw-r-- 1 voy bdrfi nst 3152 Jun 3 11:52 caMonitor.c
```

The **newepicsapp** tool creates a stub client application project with App appended to the name that you provided, and then cd's to the newly created project directory. The App suffix is enforced by EPICS – you have to learn to live with 'yourname**App**'! The stub application consists of two C language files. CaExample.c is an example of how to use ca_get(), and caMonitor.c is an example of how to setup a channel access monitor. You may do a gmake to see what happens.